

# Informática año 2000: una bomba de relojería

De verdad que no salgo de mi asombro. Los analistas internacionales del sector informático andan avisándonos muy en serio de que la cuenta atrás ha empezado, mientras que por aquí, según mis sondeos personales, casi ni hemos oído hablar del problema. Me estoy refiriendo al "problema del año 2000".

Fernando  
Sáez Vacas

Resulta que al entrar en el año 2000, muchos programas van a dedicarse inmediatamente a hacer tonterías, incapaces de distinguir en sus cálculos si las fechas que manejan pertenecen al siglo XX o al XXI. Despiste garrafal, que nos puede acarrear que un depósito bancario nuestro se transforme en incongruente saldo deudor, por ilustrar con un ejemplo relativamente trivial el conjunto de gravísimos efectos posibles sobre miles de programas y sistemas de información, y en último extremo, sobre la economía.

## EL CONVENIO DE FECHAS "MM/DD/YY", UN MAL NEGOCIO

En el fondo, toda esta miseria se debe al formato estándar de representación de las fechas: "dd/mm/aa", o "mm/dd/yy", en versión anglosajona, habitual en este mundo informatizado (d: día, day; m: mes, month; a: año; y: year). Así, mi permiso de conducir expira el 17.06.96, con lo cual el escribir este artículo me hace recordar que lo tengo que renovar enseguida, en junio de 1996. Lo mismo sucede con las tarjetas de crédito, cuya fecha de expiración se expresa por cuatro dígitos "mm/aa". Como seres humanos, manejamos de forma contextualizada estas fechas abreviadas y a mí no se me ocurriría pensar que tengo que renovar mi permiso de conducir el 17 de junio de 2096 o de 2196, épocas en las que, por lo demás, estaré previsiblemente bastante estropeado.

Pero los ordenadores ni contextualizan ni tienen sentido común, sólo poseen una referencia interna del tiempo, alguna clase de reloj o temporizador, sobre la que se efectúan los cálculos. Así, por ejemplo, si se inicia un depósito bancario en una determinada fecha del año 1995, el programa de cálculo de intereses tomará siempre el grupo 95 como sustraendo, y lo seguirá haciendo fatalmente cuando el reloj se ponga a 00, y luego a 11, etcétera, ya en pleno caos, si no se adoptan antes las medidas oportunas. ¡El ahorro de dos dígitos va a tener un coste apocalíptico!

Tanto por sus causas, como por sus consecuencias, por las estrategias de gestión a abordar, por las medidas técnicas correctoras, por las experiencias

que tengan lugar y por las lecciones que se extraigan, el problema del año 2000 pasará probablemente a la historia de la informática.

Se dice pronto: una porción del soft-

ware instalado en el mundo tiene una bomba programada para hacer explosión justo al doblar el milenio. No la ha puesto ahí ningún enemigo, sino los programadores, los técnicos de sistemas y los fabricantes de hardware y de software. ¿Por qué?

Las razones son, en primer lugar, los límites físicos de los sistemas técnicos y, después, el mal hábito de la pereza mental.

Comencemos por el principio: todo sistema tiene siempre una limitación

Utilizar dos en vez de cuatro dígitos decimales supone ahorrar dos bytes por cada dato-fecha, esto es, decenas de millares o millones de bytes de memoria si se va a procesar una base de datos

## Nunca hemos hablado de un error de aparición tan brutal como el del problema del año 2000, que rompe absolutamente todos nuestros esquemas de trabajo y curvas de coste

en cuanto a la disponibilidad de recursos. Puede ser el número de procesadores, la velocidad de los circuitos o la capacidad de memoria. Utilizar dos en vez de cuatro dígitos decimales supone ahorrar dos bytes por cada dato-fecha, esto es, decenas de millares o millones de bytes de memoria si se va a procesar una base de datos, un ahorro económico notable, no ahora, pero sí por los tiempos en los que se plantaron las semillas de este problema. Los límites físicos inevitablemente se pagan con límites en la computación. En efecto, contar los años por medio de dos dígitos supone contar por el procedimiento llamado de "módulo 100", lo que significa que el contador sólo puede contar desde 00 a 99, volviendo de nuevo a 00. Un sistema que utilice esta clase de contador nunca podrá computar centenas (siglos) ni millares (milenios).

Técnicamente, el asunto de la temporización es más complicado de lo que se acaba de exponer, debido a la idiosincrasia del reloj interno, que puede acarrear sorpresas, como la de algunos sistemas de PCs -y eso depende de las rutinas de temporización que maneje su BIOS (Basic Input/Output System)- que se reinicializarán en enero de 1980 al llegar la medianoche del 31 de diciembre de 1999.

Esta cómoda, pero mala costumbre, del convenio de representación de las fechas y de los diseños de los temporizadores se extendió bajo formas diversas, incluso cuando ya no había necesidad, al diseño de lenguajes de programación, compiladores, plataformas y aplicaciones, con el resultado de que hoy una parte del parque mundial de equipos, programas y sistemas de información está contaminado. Se conoce que todos pensaban que el mundo se acabaría antes del siglo XXI, o se creía que el ciclo de renovación completa del parque sería muy corto, o será que todos hemos practicado eso de que el que venga detrás que arree. En el sector informático siempre se ha trabajado a muy corto plazo, aunque en este asunto parecen haberse superado todas las marcas de desidia colectiva, dado que el problema se tenía detectado ya desde el decenio de los 80.

### INGENIERIA DEL SOFTWARE: LOS EQUIPOS DE CAZAFANTASMAS

Si aceptamos como hipótesis que el mundo, a pesar de todos nuestros esfuerzos, no se va a acabar antes del 2000, la conclusión es que habrá que descontaminar el parque, que es una manera eufemística de decir que habrá que desactivar la bomba. Algunos expertos estiman el coste general de tal descontaminación en cifras catastróficas alrededor de los 600.000 millones de dólares (Financial Times

Review, 7-2-1996) y avisan de que muchas empresas se quedarán fuera de los negocios gracias a este problema (o sea, que en algunos casos lo de que se acabará el mundo será metafóricamente cierto).

Especialistas del grupo Gartner estiman que reparar el software de una empresa media con unos 8000 programas infectados por el problema del tiempo puede costar entre 3,6 y 4,2 millones de dólares. Pero lo peor, nos dicen, es que se prevé que a finales de 1997 menos del 20 por ciento de las empresas habrán hecho sus deberes de descontaminación, y menos del 50 por ciento a finales de 1999. ¡Boooooom!

En los cursos de ingeniería del software hablamos de la gestión de los proyectos e insistimos en los grandes costes del mantenimiento de las aplicaciones, que oscilan entre el 50 y el 70 por ciento del total del coste de su ciclo de vida, aunque lo que se llama mantenimiento correctivo, es decir, la reparación de errores, es sólo una parte y el resto del esfuerzo se va en actualizaciones y evolución funcional de los programas. También advertimos acerca del poco entusiasmo que, pese a su trascendencia, despierta esta tarea y de lo mucho que la beneficiaría elaborar una buena documentación durante la etapa de desarrollo del software. Desde luego, nunca hemos hablado de un error de aparición tan brutal como el del problema del año 2000, que rompe absolutamente todos nuestros esquemas de trabajo y curvas de coste.

Y los rompe porque no es un error de programación, es mucho más, es un fallo de base, un fallo del sistema

industrial, es como si en un organismo vivo su sistema inmunológico se volviera loco de repente y atacara y destruyera sus propias células. La ventaja es que sabemos cual es el mal y cuando va a suceder. Sólomente, ¿sólomente?, hay que localizar los tejidos afectados y sustituirlos. Lo que nos conduce a la cuestión de descubrir cuáles son esos tejidos, cómo se localizan, cómo se sustituyen, evaluar cuánto tiempo y esfuerzo requiere la tarea y ponderar si merece la pena repararlos o, ya, metidos en gastos, cambiar la flota. (Un experto en la materia, participante en una "Conference Call" a través de Internet el 30-10-1995, cita el caso de una gran empresa de seguros con más del 90 por ciento de sus aplicaciones afectadas por el problema de las fechas).

Para responder a estas preguntas están surgiendo especialistas, conferencias, páginas en el Web de Internet y servicios de información y consultoría. Algunas empresas están desarrollando experticia (incluyendo sistemas de ayuda y herramientas de software) relativas a las numerosas facetas del problema, que afecta, no a todos, ni con la misma intensidad, pero sí a diferen-

**En el sector informático siempre se ha trabajado a muy corto plazo, aunque en este asunto del 2000 parecen haberse superado todas las marcas de desidia colectiva**

tes ordenadores y sistemas operativos, desde PCs hasta "mainframes", y a cualquier aplicación que mantenga relación algorítmica con las fechas. IBM, por ejemplo, mantiene accesible en Internet un documento de 180 páginas con pautas para comprender y planificar soluciones al problema y aprovecha para asegurar que a partir de 1996 su tecnología de PCs estará libre de esta infección.

Cabe la posibilidad de que los expertos estén interesadamente caigando las tintas para que no se me acuse de colaborar terminare relajadamente mi artículo con unas notas musicales y creando alarma social, pero, por si acaso, uno cree que lo menos aconsejable para un director de sistemas o de informática sería relegar el análisis de este problema del año 2000 para mejor ocasión. Le convendría dedicar-

se a despejar dudas y a evaluar su situación, porque no se enfrenta a un asunto meramente técnico sino a un asunto de gestión y planificación, o, más exactamente todavía, a un proyecto, que puede ser muy sencillo o muy complejo, dependiendo de los sistemas afectados, de su grado de afectación, del nivel de integración de los sistemas y de la profesionalidad de sus recursos humanos, pero probablemente ineludible.

#### FULGOR Y COCHAMBRE DE LA TECNOLOGÍA

Maravillados y sobrepasados cada día por el progreso tecnológico, ¡cuánto pasmo nos produce la aparición de estos fantasmas del pasado! Se dice, por ejemplo, que muchos de los sistemas de información afectados por el mal podrían estar corriendo sobre equipos grandes y estar escritos en Cobol, y resulta que este lenguaje, después de tantos años, sigue siendo, "lifting" tras "lifting", el más utilizado con tales equipos (Price Waterhouse Technology Forecast: 1995) y uno de los más demandados actualmente en anuncios de puestos de trabajo (p. ej., en el Reino Unido, ver Financial Times Review, 7-2-1996, pág. 2).

El problema del año 2000 es una demostración de que el pasado en informática tiene un peso formidable. Todos los días, el mercado renueva sus ofertas, mientras una parte de nuestras vidas se sigue rigiendo por sistemas antiguos, longevos y frágiles, de los que por vergüenza nunca hablamos en público, porque parecen estar negándonos el futuro.

Nuestras relaciones con la tecnología sólo pueden ser de amor y odio. Quien, sin sospecharlo para nada, mejor las expresa es el dúo de compositoras Vainica Doble con su canción "Lo eres todo", interpretada por Luz Casal (léase con música):

(...) lo eres todo, todo para mí/  
mi principio y mi fin/  
mi norte y mi guía, mi perdición/  
mi acierto y mi suerte, mi equivocación/  
eres mi suerte y mi resurrección/  
eres mi aliento y mi agonía/ (...)

(Nota: el título de esta última sección esta inspirado en la "performance" musical "El fulgor en la cochambre", de Fernando Palacios.) □